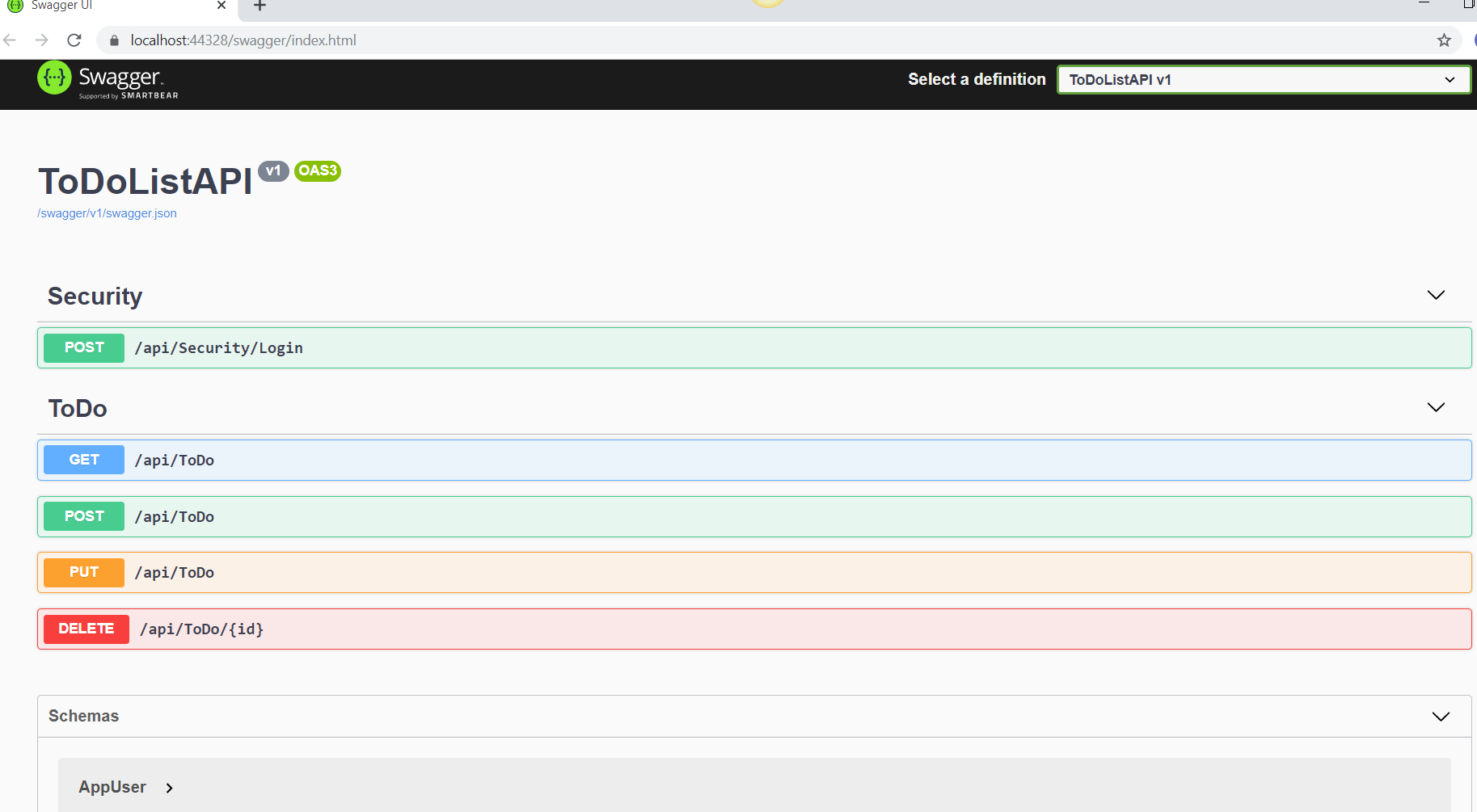
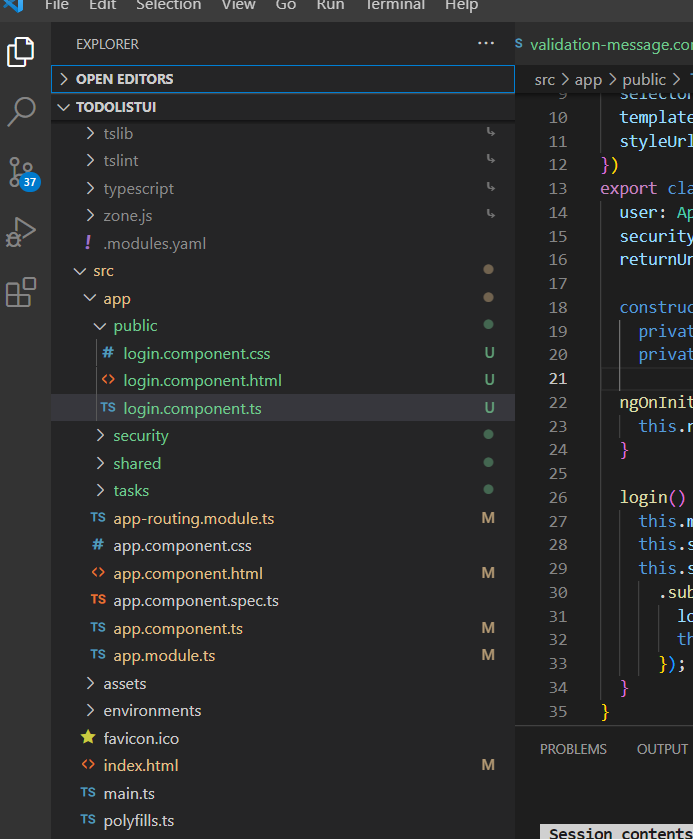
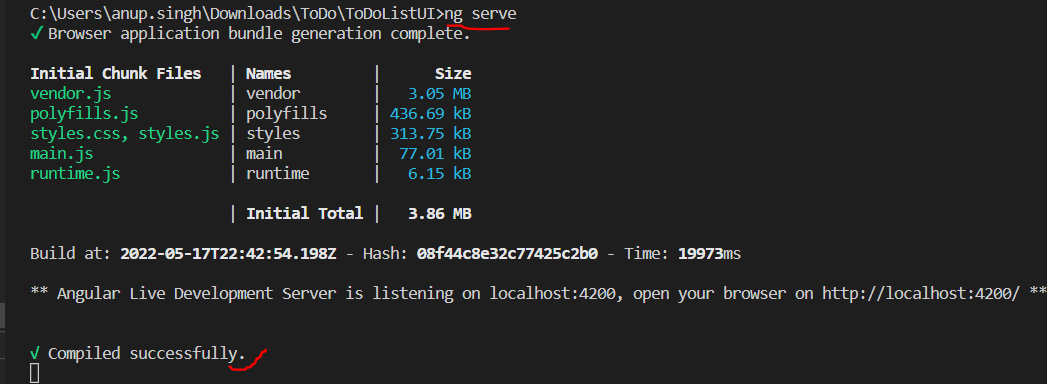
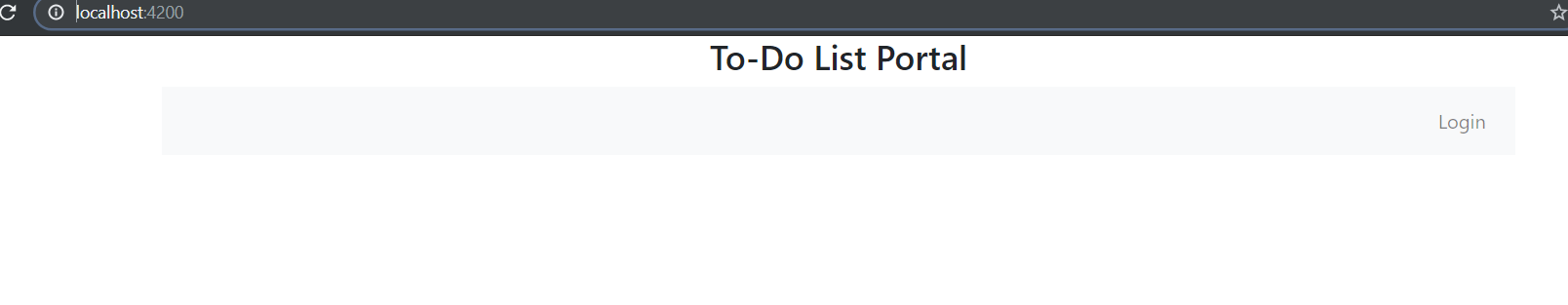
How to start the project –

1. Download the attached files.
2. It contains two folders which are the UI and API of the application.
3. Open the ‘**ToDoListAPI**’ folder in the visual studio and run it.
4. Click on the start IIS Express at the top or press F5 to run the project.
5. You can see the API working by the browser showing you the API like this –



1. Now, open the other folder ‘**ToDoListUI**’ and open it in Visual studio code. 
2. Run ng serve over the terminal and let it compile successfully.
3. On your browser run ‘<http://localhost:4200/> ' and you will be redirected to the Log-In page.



1. Run the ‘**ToDoDB.sql**’ over the SQL server to create a Database.

Approach Used-

In creating this web application, I have used Angular and Web API. The front end consists of angular and the backend consists of Web API. For the connection to the database, I used the entity framework as well as ADO.NET by dividing the project into two parts. Firstly the log-in, which includes the authentication and authorization of the user using the entity framework. Secondly, I used the ADO.NET to add, delete and edit the list of items to-do.

Why this Approach –

The above approach is implemented in my day-to-day work and I am confident that this method strengthens my work.

I worked on ASP.Net web forms which are easy to create a web application as we do not require another project for the UI. By using aspx forms I can show the front end. I opted to create this web application with the new technologies such as Web API and angular so, I can showcase what I learned in my previous job, and to be up to date I used Angular and WebAPIs.

Design Pattern Used-

I have used Dependency Injection.

What extra if get more time –

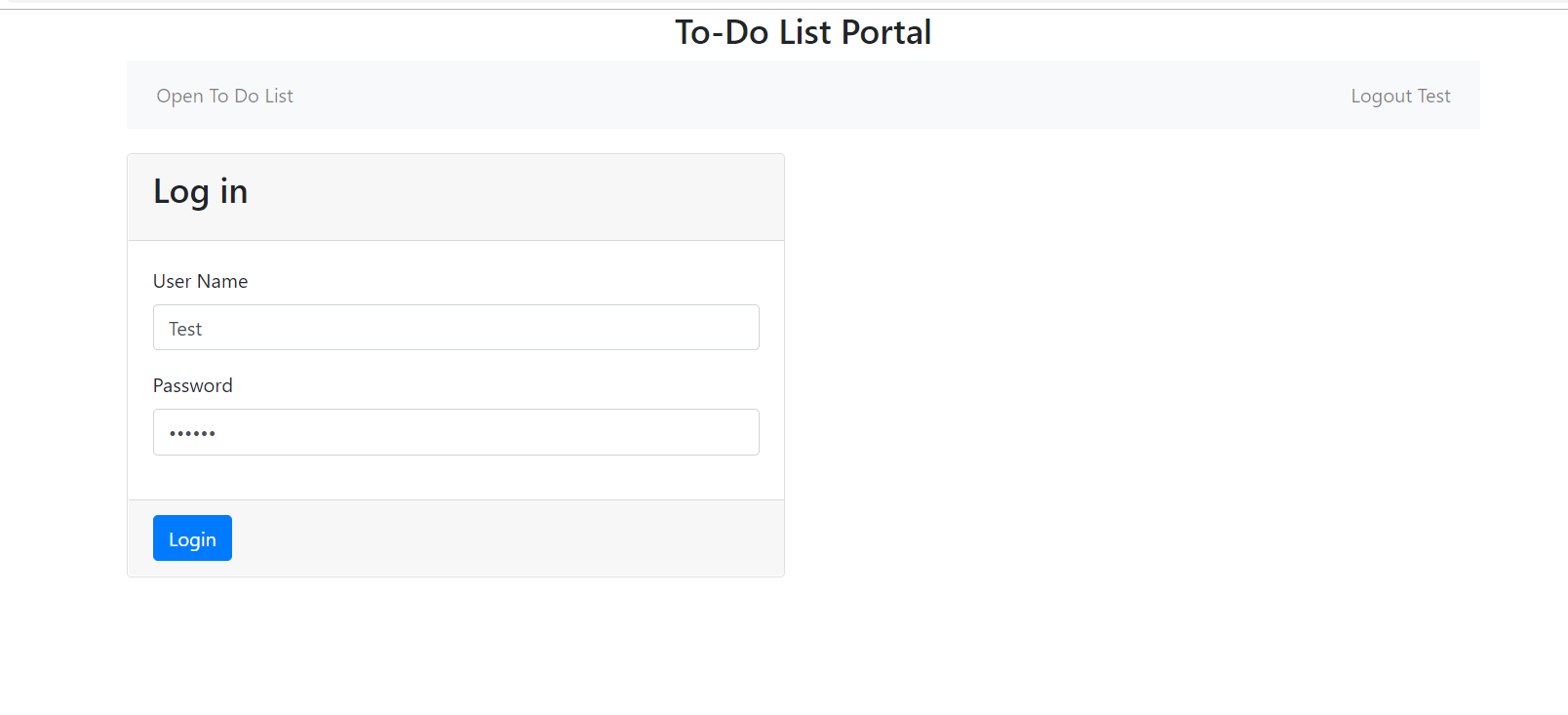
* Adding Unit test and Integration test.
* Provide UI to register a new user.
* Provide authorization over the front end where we can grant permissions to users.
* Password to be encrypted while saving into database.

Other Information –

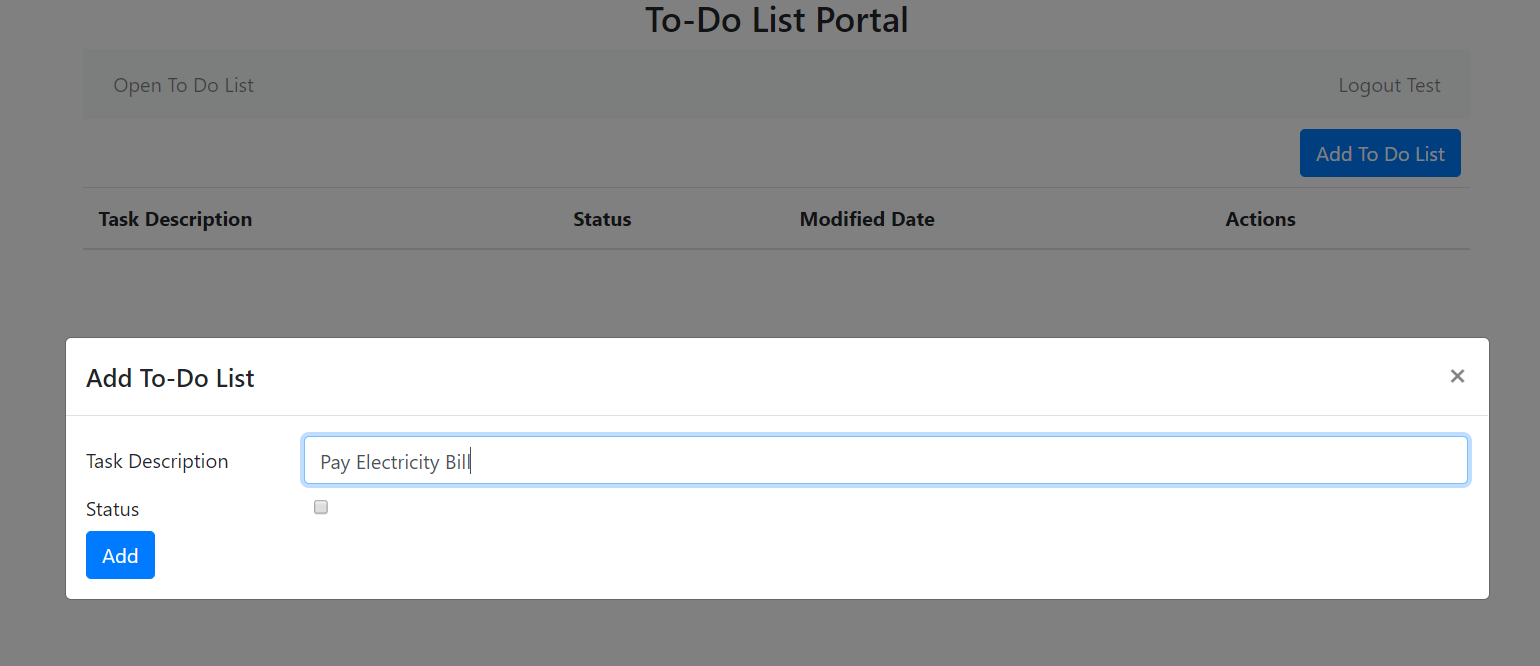
* For the security I have added the Login page where we secure the To-Do list.
* Two Users have been added where the first has all the authorization for everything and the second doesn’t have the authorization to add, delete or edit the list but can see the To-Do List.
* Routing in angular has been used to go from one page to another.
* Auth Guard is also provided for security so that using the link directly like ‘<http://localhost:4200/tasks>’ without logging in will redirect the user to the log-in page.
* To view the application, see the attached video entitled ‘Demo.mp4’.

Few Screen Shots of the application –

* Log In Page



* Adding a task



* To-Do List

